Adaptive Activation Function Generation for Artificial Neural Networks through Fuzzy Inference with Application in Grooming Text Categorisation

Zheming Zuo, Jie Li, Bo Wei, Longzhi Yang
Department of Computer and Information Sciences
Faculty of Engineering and Environment
Northumbria University, UK
Email: longzhi.yang@northumbria.ac.uk

Fei Chao
Department of Computer Science
Aberystwyth University
Aberystwyth, UK
Email: fec10@aber.ac.uk

Nitin Naik
Defence School of
Communications and Information Systems
Ministry of Defence, UK
Email: nitin.naik100@mod.gov.uk

***Abstract*—The activation function is introduced to determine the output of neural networks by mapping the resulting values of neurons into a specific range. The activation functions often suffer from 'gradient vanishing', 'non zero-centred function outputs', 'exploding gradients', and 'dead neurons', which may lead to deterioration in the classification performance. This paper proposes an activation function generation approach using the Takagi-Sugeno-Kang inference in an effort to address such challenges. In addition, the proposed method further optimises the coefficients in the activation function using the genetic algorithm such that the activation function can adapt to different applications. This approach has been applied to a digital forensics application of online grooming detection. The evaluations confirm the superiority of the proposed activation function for online grooming detection using an imbalanced data set.**

I. Introduction

The Internet has taken a large portion of youngsters' daily lives. The wide availability and rich diversity of Internet access devices make it difficult for parents and carers to monitor their kids' activity in the cyberspace. Children and young people are usually not fully aware of the risks of personal information leak, which makes them vulnerable to online groomers, without carers' attention. Paedophiles usually engage with children via social media, by building a trustful relationship [1]. This results in psychological, physical, emotional and behavioural harm [2]. Therefore, it is important to closely monitor and analyse online conversations to detect grooming activities.

Several machine learning methods have been applied to detect online child grooming with an emphasis on text feature selection from conversation records, such as a chat logs classification system using support vector machine (SVM) as reported in the work of [3]. Online child grooming conversation text categorisation is a typical example of the class-imbalanced research problem, where such texts highly vary in duration, type, and intensity depending on the perpetrator characteristics and behaviour [4]. Unfortunately, most of the conventional machine learning algorithms cannot yield good performance on imbalanced datasets [5]. Therefore, the performance of the online child grooming detection using conventional machine learning approaches needs to be improved.

The intuitive solution is to artificially balance the imbalanced dataset. It can be achieved by either over-sampling data instances of the minority class or down-sampling data samples of the majority class [6]. However, over-sampling the minority of data samples may lead to overfitting due to the duplication of data instances. Similarly, the method of down-sampling the majority can cause information loss, as the most important data samples may fail to be selected during the process of the down-sampling [6].

This paper proposes an adaptive fuzzy inference-based activation function generation approach for artificial neural networks (ANNs) to address the above issue. Conventional fuzzy inference maps a given input to an output based on a dense rule base, whilst fuzzy rule interpolation enhances the conventional fuzzy inference to work also with sparse rule bases [7], [8]. Both approaches have been extended to support the adaptation of fuzzy rule bases for various applications [9]–[12], including those with imbalanced datasets as commonly seen in online grooming detection.

The proposed approach takes the advantages of adaptive fuzzy inference. It first generates a rule base regarding a given application to represent a bespoke activation function, which is then optimised using a genetic algorithm (GA) such that the supported ANN is adapted to the particular application. This paper also applied the proposed approach to a highly imbalanced grooming detection data set. The experimental results confirm that the efficacy of the proposed approach in handling the common issues of conventional activation functions and imbalance datasets.

The rest of the paper is organised as follows. Section II introduces the existing activation functions. Section III presents the proposed AdaTSK activation function. Section IV details the experimental results for comparison and validation. Section V concludes the paper.

II. Background

Activation functions are of crucial importance in ANNs and deep neural networks (DNNs) [13]–[18]. Basically, in ANNs and DNNs, the output from inputs is defined by one activation function, which is essentially a mechanism to distinguish the information about the relevance of a neuron. It is worthwhile to note that the activation function needs to be continuous and differentiable. The general form of an activation function can

be represented as:

$$f(x) = \sum(w * x) + b, \tag{1}$$

where x and w respectively denotes the input(s) to a neuron and the weight, and b is the associated bias.

A list of commonly used activation functions are summarised in Table I. In particular, Sigmoid is still the most common activation function in the ANNs due to the easiness of derivation calculation, but it suffers from 'gradient vanishing' and 'zero centred' issues. To solve these issues, TanH (*a.k.a.* hyperbolic tangent) activation function, is used for generating 'zero centred function outputs' as well as solving the 'gradient vanishing'. Recently, the rectified linear unit (ReLU) [13] activation function is proposed to solve the 'gradient vanishing' issue and accelerate calculation as well as the convergence rate.

TABLE I. A SUMMARY OF EXISTING ACTIVATION FUNCTIONS.

Activation Function	Formulation
Identity	$f(x) = x$
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
TanH	$f(x) = \frac{(e^{x}-e^{-x})}{(e^{x}+e^{-x})}$
ArcTan	$f(x) = tan^{-1}(x)$
Sinusoid [14]	$f(x) = \sin(x)$
Gaussian	$f(x) = e^{-x^2}$
SoftPlus [16]	$f(x) = \ln(1 + e^{x})$
ReLU [13]	$f(x) = \max(0, x)$
Leaky ReLU [17]	$f(x) = \max(\alpha x, x), \alpha = 0.01$
ELU [18]	$f(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha(e^{x} - 1), & \text{otherwise.} \end{cases}$
SiLU [15]	$f(x) = x \cdot \sigma(x)$

The ReLU activation function still, unfortunately, has the 'dying ReLU' problem, which results in a proportion of neurons are never activated, and their corresponding parameters cannot be updated. Possible solutions include the Xavier initialisation method [19] and adjusting reasonable learning rate. The leaky ReLU [17] and exponential linear units (ELU) [18] were proposed to address the 'dying ReLU' and solve the 'zero centred function outputs' issues with the additional computational cost required by the latter one. Most recently, a sigmoid-weighted linear unit (SiLU) [15] is constructed by computing the sigmoid function multiplied by its input, which has proven a well-formulated activation function for function approximation in the domain of reinforcement learning.

III. PROPOSED ACTIVATION FUNCTION GENERATION

The proposed inference-based activation function generation approach for ANNs, particularly the back-propagation neural networks (BPNN), is detailed in this section. The structure of interactions between BPNN and the proposed activation function generation is illustrated in Fig. 1. Concretely, TSK fuzzy inference system provides the baseline of the process of the GA. An existing activation function, Sigmoid or TanH function, is simulated by employing the TSK fuzzy inference system. Then, the GA is employed to fine-tune the parameters of the initialised TSK-based activation function in order to select the optimal activation function for the particular problem

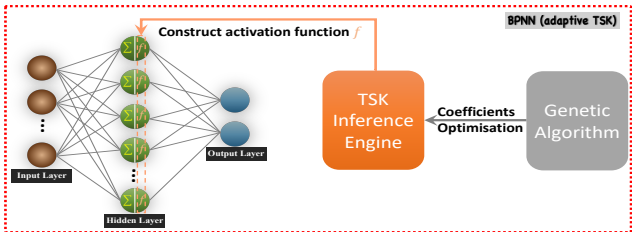


Fig. 1. The proposed adaptive TSK activation function generation approach

and enhance the performance. Each component is detailed in the following subsections.

A. TSK Model

The TSK fuzzy model [20] is a typical fuzzy rule for the TSK model is of the following form:

$$\textbf{IF } x \textit{ is } A \textbf{ THEN } w = f(x), \tag{2}$$

where A is fuzzy sets regarding antecedent variables x, and $f(x)$ is a crisp function (usually polynomial), which determines the crisp value of the consequent. Assume that a rule base for the TSK fuzzy model is comprised of n rules:

$$R_i: \textbf{ IF } x \textit{ is } A_i \textbf{ THEN } z = f_i(x) = a_i x + b_i \quad \forall i \in [1, n], \tag{3}$$

where a_i, b_i are constants in the polynomial equation in the consequent part of the rule. Given an observation with the singleton value as input (x), the final inference result can be obtained by:

$$f(x) = \frac{\sum_{j=1}^{i} \alpha_j * \Phi_j(x)}{\sum_{j=1}^{i} \alpha_j}, \tag{4}$$

where α_j represents the firing strength of the j^{th} rule, and $\Phi_j(x)$ denotes the corresponding intermediate result of j^{th} rule, subject to the given input x. It has been proven in [21] that different types of membership functions do not introduce different inference results if they are properly fine-tuned. Thus, the Gaussian membership function is used in this work for computational efficiency, represented by

$$A_i = \exp\left(-\frac{1}{2}\left(\frac{x - c_i}{\sigma_i}\right)\right), \tag{5}$$

where c_i is the location of the centre of the peak, and σ_i represents the width of the bell-shaped curve. More details about TSK fuzzy model can be found in [20].

B. Initial TSK Rule Base Generation

As discussed in Section II, a number of activation functions have been proposed. The selection of an activation function is highly problem-dependent to the distribution of the extracted features. In order to accelerate the training process, a fast approximation function is always selected. In this work, a single input and single output first-order type-1 TSK fuzzy system has been initialised which aims to simulate an existing mathematically formulated activation function to provide the baseline of the process of the population generation during the GA. The processes of the TSK rule base initialisation is summarised as follows:

Step 1 - Input domain range determination: Given an input domain, the first step of modelling is to determine its range. In this work, it can be obtained by determining the actual range of the output of each neuron in the hidden layers. Assume that an ANN model contains m neurons in its hidden layers, the range of the input domain can be obtained by: $[\underline{x}, \bar{x}] = \left[\min\left(\sum_{j=1}^{i} w_j^m a_j^m\right), \max\left(\sum_{j=1}^{i} w_j^m a_j^m\right)\right]$, where w_j^m is the weight of j^{th} input of m^{th} neuron, and i represents the total number of inputs for m^{th} neuron.

Step 2 - Domain partition: The consequence of the first order type-1 TSK fuzzy rule is a linear function. Therefore, the problem domain is partitioned into a ($a \in \mathbb{N}$) grid areas by the piecewise linear technique. Each obtained sub-area can be expressed as a linear function. Fig. 2 shows the example of the approximated linear functions in the Sigmoid function where three sub-regions are partitioned.

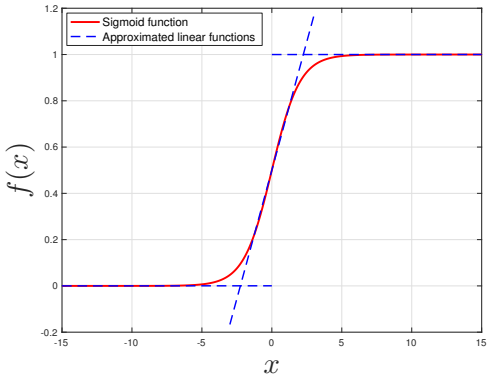


Fig. 2. Piecewise linear approximation for the Sigmoid function

Step 3 - TSK rule base initialisation: Each sub-region is represented by one TSK fuzzy rule. The rule antecedents, the Gaussian membership functions, is obtained by a fuzzy partition of the input domain of the corresponding sub-area. The rule consequence is the determined linear function of this particular sub-area from *Step 2*. The initial TSK rule base can then be constructed by combining all the formulated rules.

C. TSK Rule Base Optimisation

The final activation function is generated by fine-tuning the rule base using the general optimisation searching algorithm, GA. Briefly, GA is an adaptive heuristic search algorithm for solving both constrained and unconstrained optimisation problems based on evolutionary algorithms (EAs), which has been widely employed in the rule bases optimisation, such as [22], [23] and [24]. The algorithm starts from an initialised population to select the required number of individuals for reproduction by applying the genetic operators: crossover and mutation. The offspring and some of the existing individuals jointly produce the next generation of the population. The algorithm repeats such process until the satisfactory solution is obtained or a maximum number of iterations has been reached. Details of the GA is explained below.

1) Problem Representation: In the GA, an individual or chromosome, denoted as I, is used to represent a candidate solution. In this work, an individual is designed to represent an entire rule base. Assume that a TSK rule base is comprised of n rules as defined in Eq. (3), an individual is then formed as illustrated in Fig. 3.

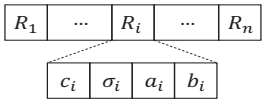


Fig. 3. The chromosome encoding

2) Population Initialisation: The initial population $\mathbb{P} = \{I_1, I_2, \cdots, I_{|\mathbb{P}|}\}$ is constructed by the introduced initialised rule base and its random variations. The size of the population $|\mathbb{P}|$ is typically with a range from 20 to 30 [23], depending on one specific problem.

3) Objective Function: An objective function is used in GA to assess the quality of individuals. The objective function, in this work, is defined as the system prediction accuracy from the k-fold cross-validation mechanism. Given a training data set $\mathbb{T}$ and an individual I_i, $1 \leqslant i \leqslant |\mathbb{P}|$, the k-fold cross-validation prediction accuracy (CV) can be calculated as:

$$CV_k(I_i) = 1 - \frac{1}{k}\sum_{j=1}^{k} E_j(I_i), \tag{6}$$

where k is the number of folds of the training data set $\mathbb{T}$, and $E_j(I_i)$ is the system prediction error for j^{th} fold, which can be obtained by:

$$E_j(I_i) = \frac{\text{Number of misclassified samples}}{\text{Total number of samples}}. \tag{7}$$

The individual with the best accuracy is selected as the solution within the population.

4) Individual Selection: The fitness proportionate selection method (*a.k.a.* 'roulette wheel selection') is employed in this work to select the required number of individuals for reproduction.

5) Individual Reproduction: Given a number of selected individuals (termed as parents), the genetic operators, crossover and mutation, are used in the next generation of the population. In this work, we use the single-point crossover approach to swap all data beyond a swapping point between two selected individuals. And the uniform mutation approach, which replaces the value of the selected gene with a random value selected between the specified upper and lower bounds, is adopted. Note that the upper and lower bounds have been previously determined in Section III-B. The crossover and mutation rate were set to 0.9 and 0.05 in this work. To ensure each bred individual is a valid solution as well as satisfy the requirement of the condition of the proposed activation function, the following constraints have been applied during the process of the reproduction.

a) For solving the 'dead neuron' problem, Eq. (8) has to be satisfied to ensure that the weight of each neuron can be updated for each iteration during the reproduction process.

$$\exists f\left(\sum_i\right) \neq 0 \tag{8}$$

Fig. 4. Pipeline for online grooming detection.

b) To solve the 'non zero-centred' problem, Eq. (9) has been used as one of the constraints.

$$\begin{aligned} \forall f\Big(\sum_i (w_i x_i + b)\Big) &\gtrless 0, \\ \forall f\Big(\sum_i (w_i x_i + b)\Big) &\lessgtr 0. \end{aligned} \tag{9}$$

c) To mitigate the 'gradient vanishing' and 'gradient blowing up' issues, the Eq. (10) is checked during the reproduction process to determine the gradient relationship between two layers during the learning process and guarantee the corresponding gradient relationships fall in the given range.

$$C_{min} \leqslant \sum_{l_{m-n+1}=1}^{v} \cdots \sum_{l_m=1}^{v} \prod_{s=1}^{n} w_{l_s l_{s-1}} f'_{l_s}(m-s) \leqslant C_{max}, \tag{10}$$

where l_{m-n+1} represents the $(l_{m-n+1})^{th}$ node (*i.e.* neuron) in $(m-n+1)^{th}$ layer, $w_{l_s l_{s-1}}$ is the weight between ${l_s}^{th}$ node in s^{th} layer and $(l_{s-1})^{th}$ node in $(s-1)^{th}$ layer, and $f_{l_s}(t)$ denotes the output of ${l_s}^{th}$ node in t^{th} layer, the $f'_{l_s}(\cdot)$ is the partial derivative of $f_{l_s}(\cdot)$. C_{min} and C_{max} are the given threshold, which indicates the range of the valid gradient relationship between two layers. Note that the 'gradient vanishing' and 'blowing up' issues could not be completely resolved at the same time. Thus, this constraint is applied trying to suppress (or relieve) the issues as much as possible.

6) Iteration and Termination: The selection and reproduction processes are iterated until that (1) the value of individual in the objective function is less than a pre-specified threshold; or (2) the pre-defined number of maximum iterations is reached. When the stopping criteria is satisfied, the fittest individual in the current population is deemed as the solution.

IV. Experiments

The proposed adaptive TSK-based activation function was applied in the BPNN, which is validated and evaluated using a reconstructed highly class-imbalanced grooming dataset [2]. In addition, a comparative study was conducted in investigating the performance of the proposed activation function in reference to 11 classic ones. All the experiments were conducted on the basis of the pipeline that visualised in Fig. 4.

A. The Data Set

The data set contains 1,000 XML files for binary and multi-label (*i.e.* 3) classification tasks with a wide diversity of chatting contents. For binary classification, the number of documents associated with label 'Normal' and 'Abnormal' is 731 and 269, respectively. For multi-label classification, the number of documents for 'Normal' is constant whereas the rest 236 and 33 files are from the 'Abnormal' class, which has been further divided into 'Pedophile' and 'Sex' categories. To further present the degree of imbalance of this dataset, the feature embedding of the BoW and TFIDF features were visualised in Fig. 5, which obviously reveals the difficulty of classification. Concretely, based on the final optimal results (except AdaTSK) summarised in Fig. 8, the best performance for the rest eleven activation function is around 73.10% (for binary classification) or 73.11% (for multi-label classification), in which the only exception is the 73.20% that achieved by ReLU in the case of binary classification using the TFIDF features.

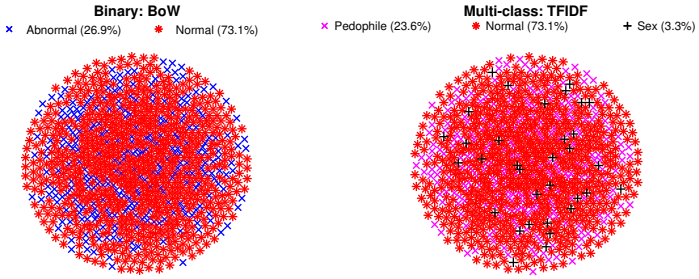


Fig. 5. Feature embedding visualisations of the BoW and TFIDF features on the experimented dataset using t-SNE [25]. Each document is visualised as a point and documents belonging to the same category have the same colour.

B. Experimental Setup

The experiments are implemented in Python™ 2.7.14 and conducted using a workstation equipped with AMD® Ryzen™ Threadripper™ 1950X (16-core) CPU @ 3.40 GHz and 64GB RAM.

For extracting text features, the bag of words (BoW) and term frequency-inverse document frequency (TFIDF) were used [2]. The extracted feature dimension was fixed to 80. This was followed by selecting 50 attributes only from the extracted 80 attributes using the fuzzy-rough feature selection (FRFS) approach ([26]). For normalising the selected features, as evaluated by [2], existing techniques, such as min-max (MM) normalisation, ℓ_1-normalisation, ℓ_2-normalisation, power normalisation (PN), and their variants (*i.e.* ℓ_1PN, ℓ_2PN, PNℓ_1, and PNℓ_2), were employed for yielding comparative classification performance for BPNN [27]–[29]. Additionally, the power coefficient α employed in PN and its variants were valued as 0.1.

When setting up the BPNN classifier [27]–[29], the initialised learning rate, momentum and regularisation (or penalty) term were set to 0.1, 0.9 and 0.0001, and the number of neurons in the hidden layer was set to 25. For optimising the performance of BPNN, the stochastic gradient descent (SGD)

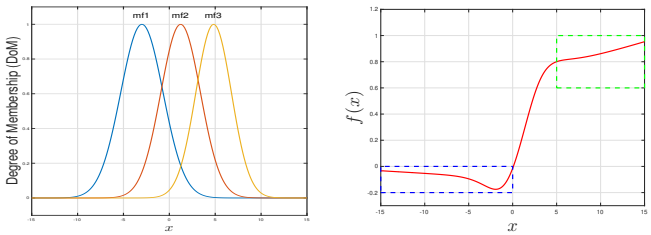
and GA were respectively employed in using 11 existing activation functions and the proposed one. 10-Fold cross-validation was employed in the testing phase.

C. Generated and Optimised Activation Function

The optimised TSK rule base has been listed in Table II and depicted in Fig. 6.

TABLE II. OPTIMISED TSK RULE BASE

No. i	Input A_i	Consequence $f_i(x)$
1	$c_1 = -2.99, \sigma_1 = 2.31$	$f_1(x) = -0.004x + -0.093$
2	$c_2 = 31.23, \sigma_1 = 2.16$	$f_2(x) = 0.21x + -0.01$
3	$c_3 = 4.83, \sigma_1 = 1.94$	$f_3(x) = 0.02x + 0.65$



(a) (b)

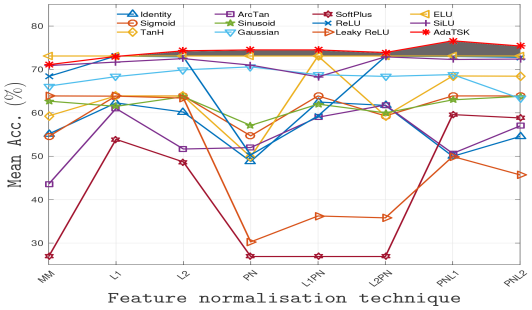
Fig. 6. The (a) input and (b) output of the optimised AdaTSK activation function.

Noteworthy, in Fig. 6(b), the optimised AdaTSK activation function possesses four advantages: (1) eliminating the hidden damages of 'dead neurons' ($f(x) \neq 0, \forall x < 0$, refer to the blue dotted region); (2) generating the 'zero-centred function outputs' (the gradients can be updated in both positive and negative directions); (3) suppressing the 'gradient vanishing' (*i.e.* the value of $f(x)$ is kept increasing when x is increased, see the green dotted region); (4) relieve the 'exploding gradients' since, with the fuzzy rule constraint, $x \in [-15, +15]$ rather than increased to $+\infty$ or decreased to $-\infty$.

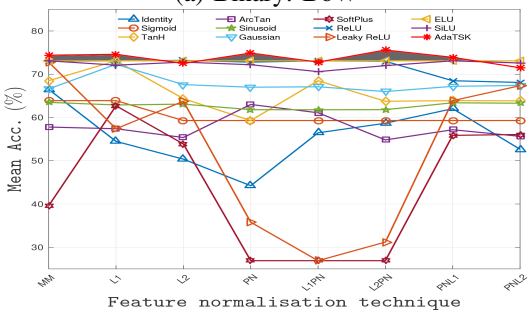
D. Results Analysis and Discussion

Fig. 7 shows the accuracy of 12 activation functions, and Fig. 8 summarises the best performance. Then, the best accuracies obtained using AdaTSK were used to compare with those shown in the existing work [2] of grooming detection.

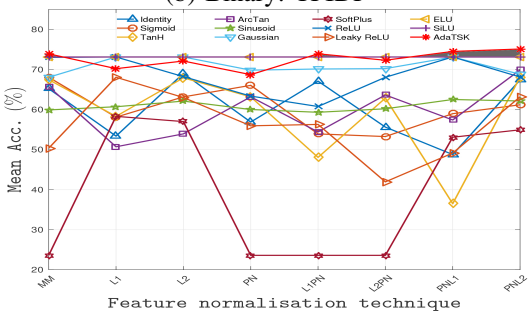
Fig. 7 and 8 show AdaTSK yields the best performance in binary and multi-label classifications using BoW and TFIDF features. When using BoW features for binary classification as shown in Fig. 7(a), AdaTSK achieves 76.60% using PNℓ_1 whereas TanH (using ℓ_1PN), ReLU (using ℓ_1), and ELU (the performance is consistent when using different normalisation techniques) produces 73.10%. Fig. 7(b) shows TFIDF features in binary classification. The peak performance of AdaTSK is 75.60% (with ℓ_2PN), which is followed by 73.20% that yielded by ReLU (with ℓ_2), while TanH (with ℓ_1), ELU (no change obtained when using different normalisation approaches), and SiLU (with MM and PNℓ_1) all achieve 73.10%. In the scenario of BoW features to be classified into multiple labels as presented in Fig. 7(c), the best performance of AdaTSK is 75.10% (with PNℓ_2 applied), the second best accuracy is 73.11% when using Gaussian (with ℓ_1, ℓ_2, and PNℓ_1), ReLU (with ℓ_1 and PNℓ_1). ELU and SiLU are consistently the same



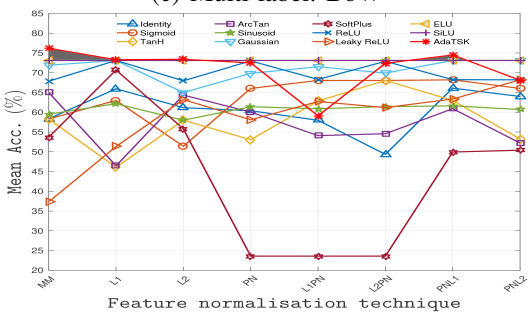
(a) Binary: BoW



(b) Binary: TFIDF



(c) Multi-label: BoW

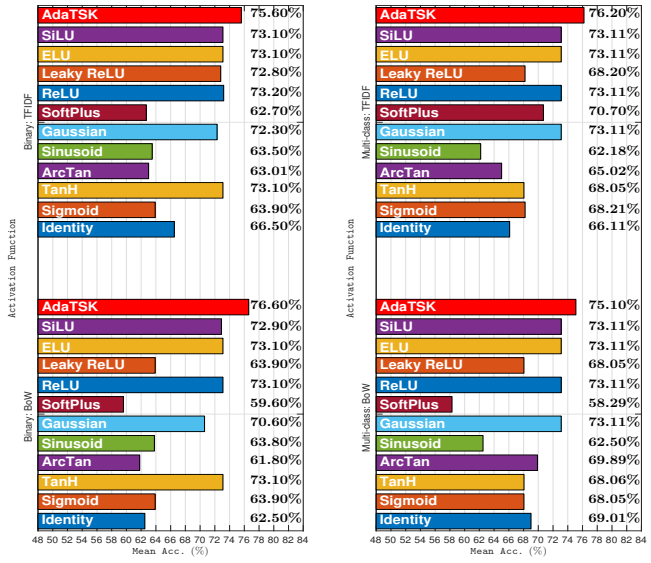


(d) Multi-label: TFIDF

Fig. 7. Classification accuracies of different activation functions by varying the feature normalisation techniques. The grey region demonstrates the difference between the proposed AdaTSK and 11 different activation functions adopted by BPNN for binary and multi-label classification tasks. Best viewed in color.

when different normalisation techniques applied. When using the TFIDF features for multi-label classification (shown in Fig. 7(d)), AdaTSK reached its best performance of 76.20% using MM normalisation technique, while Gaussian (when ℓ_1, PNℓ_1, and PNℓ_2 adopted), ReLU (when ℓ_1 is used), ELU (when all the normalisation techniques used), and SiLU (using any of the eight normalisation techniques is employed) all generate 73.11% accuracy.

To conclude, contributed by four superior properties (Sec. IV-C), the proposed AdaTSK activation function has better classification performance for the highly unbalanced dataset.



(a) Binary (b) Multi-class

Fig. 8. Performance summary by varying the feature normalisation techniques.

TABLE III. A COMPARISON STUDY

Task Type	Method	Feat. Dim.	Acc.
Binary	TFIDF + PNℓ2 + Logistic Regression [2]	120	73.21%
	BoW + PNℓ1 + BPNN (AdaTSK)	**50**	**76.60%**
Multi-label	BoW + PNℓ2 + Logistic Regression [2]	180	73.30%
	TFIDF + MM + BPNN (AdaTSK)	**50**	**76.20%**

V. CONCLUSION

An adaptive activation function using adaptive fuzzy inference has been proposed in this paper to improve the performance of BPNN on imbalanced grooming detection dataset, in addition to addressing the issues common to other existing activation functions. Though promising experimental results have been obtained, more real-world datasets are worth trying as part of the future work. Also, other adaptive fuzzy inference approaches need to be investigated to better support the proposed activation generation approach.

REFERENCES

[1] A. E. Cano, M. Fernandez, and H. Alani, "Detecting child grooming behaviour patterns on social media," in *Proc. Int. Conf. Social Informat.*, 2014, pp. 412–427.

[2] Z. Zuo, J. Li, P. Anderson, L. Yang, and N. Naik, "Grooming detection using fuzzy-rough feature selection and text classification," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2018, pp. 1–8.

[3] S. J. Pandey, I. Klapaftis, and S. Manandhar, "Detecting predatory behaviour from online textual chats," in *Proc. Int. Conf. Multimedia Commun. Services and Security*, 2012, pp. 270–281.

[4] F. E. Gunawan, L. Ashianti, and N. Sekishita, "A simple classifier for detecting online child grooming conversation." *Telkomnika*, vol. 16, no. 3, pp. 1239–1248, 2018.

[5] Y. B. Wah, H. A. A. Rahman, H. He, and A. Bulgiba, "Handling imbalanced dataset using svm and k-nn approach," in *Proc. AIP Conf.*, vol. 1750, no. 1, 2016, pp. 020 023.1–020 023.8.

[6] S. Kotsiantis, D. Kanellopoulos, P. Pintelas *et al.*, "Handling imbalanced datasets: A review," *GESTS Int. Trans. Comput. Sci. Eng.*, vol. 30, no. 1, pp. 25–36, 2006.

[7] L. Yang, Z. Zuo, F. Chao, and Y. Qu, "Fuzzy interpolation systems and applications," in *Modern Fuzzy Control Systems and Its Applications*, S. Ramakrishnan, Ed. Rijeka: IntechOpen, 2017.

[8] L. Yang and Q. Shen, "Closed form fuzzy interpolation," *Fuzzy Sets and Syst.*, vol. 225, pp. 1–22, 2013, theme: Fuzzy Systems.

[9] H.-J. Rong, N. Sundararajan, G.-B. Huang, and G.-S. Zhao, "Extended sequential adaptive fuzzy inference system for classification problems," *Evolving Systems*, vol. 2, no. 2, pp. 71–82, Jun 2011.

[10] L. Yang and Q. Shen, "Adaptive fuzzy interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 6, pp. 1107–1126, 2011.

[11] L. Yang, F. Chao, and Q. Shen, "Generalized adaptive fuzzy rule interpolation," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 4, pp. 839–853, 2017.

[12] J. Li, H. P. H. Shum, X. Fu, G. Sexton, and L. Yang, "Experience-based rule base generation and adaptation for fuzzy interpolation," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2016, pp. 102–109.

[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[14] M. S. Gashler and S. C. Ashmore, "Training deep fourier neural networks to fit time-series data," in *Proc. Int. Conf. Intell. Comput.*, 2014, pp. 48–55.

[15] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[16] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Int. Conf. Artif. Intell. Stats.*, 2011, pp. 315–323.

[17] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 3.1–3.6.

[18] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–14.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stats.*, 2010, pp. 249–256.

[20] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 15, no. 1, pp. 116–132, Jan 1985.

[21] P.-C. Chang and C.-Y. Fan, "A hybrid system integrating a wavelet and tsk fuzzy rules for stock price forecasting," *IEEE Trans. on Syst., Man, and Cybern. C, Appl. Rev.*, vol. 38, no. 6, pp. 802–815, 2008.

[22] N. Elisa, J. Li, Z. Zuo, and L. Yang, "Dendritic cell algorithm with fuzzy inference system for input signal generation," in *Proc. UK Work. Comput. Intell.*, 2018, pp. 203–214.

[23] J. Li, L. Yang, Y. Qu, and G. Sexton, "An extended takagi–sugeno–kang inference system (tsk+) with fuzzy interpolation and its rule base generation," *Soft Computing*, vol. 22, no. 10, pp. 3155–3170, 2018.

[24] Y. Tan, J. Li, M. Wonders, F. Chao, H. P. Shum, and L. Yang, "Towards sparse rule base generation for fuzzy rule interpolation," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2016, pp. 110–117.

[25] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[26] Q. Guo, Y. Qu, A. Deng, and L. Yang, "A new fuzzy-rough feature selection algorithm for mammographic risk analysis," in *Proc. Int. Conf. Natrl. Comput. Fuzzy Syst. Knowl. Discovery*, Aug 2016, pp. 934–939.

[27] Z. Zuo, L. Yang, Y. Peng, F. Chao, and Y. Qu, "Gaze-informed egocentric action recognition for memory aid systems," *IEEE Access*, vol. 6, pp. 12 894–12 904, 2018.

[28] Z. Zuo, D. Organisciak, H. P. H. Shum, and L. Yang, "Saliency-informed spatio-temporal vector of locally aggregated descriptors and fisher vectors for visual action recognition," in *Proc. British Mach. Vis. Conf.*, 2018, pp. 321.1–321.11.

[29] Z. Zuo, B. Wei, F. Chao, Y. Qu, Y. Peng, and L. Yang, "Enhanced gradient-based local feature descriptors by saliency map for egocentric action recognition," *Appl. Syst. Innov.*, vol. 2, no. 1, 7, Jan 2019.